## AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

### Listing of Claims:

1.     (Currently Amended) In a networked environment, wherein one or more client computer systems make requests for information from a server computer system, the server computer system providing information in response to the requests from the one or more client computer systems, the server computer system having one or more listen sockets and having a backlog queue for queuing connection requests that the server computer system cannot currently handle, a method of the server computer system reducing denials of service even though the server computer system is experiencing a denial of service attack, the method comprising:

receiving a plurality of connection requests from said one or more client computer systems;

establishing a connection socket for at least one of the plurality of connection requests without placing the connection request in a backlog queue;

for each connection request for which the server computer system cannot currently establish a connection socket, placing the connection request in the backlog queue without then establishing a connection socket, wherein the backlog queue is capable of containing connection requests that include associated request data and connection requests that do not include associated request data;

determining that the backlog queue is being used;

in response to the determination, identifying any connection sockets that have no received request data; and

disconnecting the identified connection sockets, so as to reduce an impact of a denial of service attack.

2.     (Original)     The method in accordance with Claim 1, further comprising mapping each connection request to a corresponding listen socket.

3.   (Original)   The method in accordance with Claim 2, wherein each listen socket has a corresponding backlog queue.

4.   (Original)   The method in accordance with Claim 3, wherein placing the connection request in a backlog queue comprises placing the request in the backlog queue corresponding to the listen socket that the connection request mapped to.

5.   (Previously Presented) The method in accordance with Claim 1, wherein establishing a connection socket for at least one of the plurality of connection requests is performed using a Winsock module.

6.   (Previously Presented) The method in accordance with Claim 1, wherein establishing a connection socket for at least one of the plurality of connection requests comprises calling a module that accepts connections and waits for associated request data before completing.

7.   (Previously Presented) The method in accordance with Claim 6, wherein the module that accepts connects and waits for associated request data before completing comprises a Winsock()AcceptEx() module.

8.   (Previously Presented) The method in accordance with Claim 1, further comprising monitoring the backlog queue, the determination being made while monitoring the backlog queue.

9.   (Previously Presented) The method in accordance with Claim 12, wherein determining that the backlog queue is being used comprises detecting that the module that scans at least the backlog queue has returned.

10.     (Previously Presented) The method in accordance with Claim 12, wherein the module that scans at least the backlog queue for activity comprises a Winsock()select() module.

11.     (Original) The method in accordance with Claim 10, wherein determining that the backlog queue is being used comprises detecting that the Winsock()select() module has returned.

12.     (Previously Presented) The method in accordance with Claim 8, wherein monitoring the backlog queue comprises calling a module that scans at least the backlog queue for activity.

13.     (Currently Amended) The method in accordance with Claim ~~1~~30, wherein identifying any connection sockets that have connections but no received request data comprises the following:

        calling a module that identifies the state of the connection socket.

14.     (Original) The method in accordance with Claim 13, wherein the module that identifies the state of the connection socket comprises a Winsock()getsockopt() module.

15.     (Previously Presented) The method in accordance with Claim 1, further comprising:

        specifying a grace period spanning the time the backlog queue is determined to be used and the time the identified sockets are disconnected, wherein the disconnection is performed only if the backlog queue still has entries after the grace period.

16.     (Cancelled)

17.     (Currently Amended) A computer program product for use in a networked environment, wherein one or more client computer systems make requests for information from a server computer system, the server computer system providing information in response to the requests from the one or more client computer systems, the server computer system having one or more listen sockets and having a backlog queue for queuing connection requests that the server computer system cannot currently handle, a computer program product for implementing a method of the server computer system reducing denials of service even though the server computer system is experiencing a denial of service attack, wherein the computer program product comprises computer-executable instructions which, when executed by a processor, implements the following:

receiving a plurality of connection requests from said one or more client computer systems;

establishing a connection socket for at least one of the plurality of connection requests without placing the connection request in a backlog queue;

for each connection request for which the server computer system cannot currently establish a connection socket, placing the connection request in a backlog queue without then establishing a connection socket, ~~wherein the backlog queue is capable of containing connection requests that include associated request data and connection requests that do not include associated request data~~;

determining that the backlog queue is being used;

in response to the determination, identifying any connection sockets that have no received request data; and

disconnecting the identified connection sockets, <u>so as to reduce an impact of a denial of service attack</u>.

18.     (Original) The computer program product in accordance with Claim 17, further comprising computer-executable instructions for mapping each connection request to a corresponding listen socket, wherein each listen socket has a corresponding backlog queue.

19.    (Original) The computer program product in accordance with Claim 17, wherein the computer-executable instructions for placing the connection request in a backlog queue comprise computer-executable instructions for placing the request in the backlog queue corresponding to the listen socket that the connection request mapped to.

20.    (Previously Presented) The computer program product in accordance with Claim 17, wherein the computer-executable instructions for establishing a connection socket for at least one of the plurality of connection requests comprises at least portions of a Winsock module.

21.    (Cancelled)

22.    (Previously Presented) The computer program product in accordance with Claim 17, further comprising computer-executable instructions for performing the following:
specifying a grace period spanning the time the backlog queue is determined to be used and the time the identified sockets are disconnected, wherein the disconnection is performed only if the backlog queue still has entries after the grace period.

23-25.    (Cancelled)

26.     (New) In a networked environment, wherein one or more client computer systems make requests for information from a server computer system, the server computer system providing information in response to the requests from the one or more client computer systems, the server computer system having one or more listen sockets and having a backlog queue for queuing connection requests that the server computer system cannot currently handle, a method of the server computer system reducing denials of service even though the server computer system is experiencing a denial of service attack, the method comprising:

receiving a denial of service attack comprising a plurality of connection requests from a client computer system without receiving any associated request data for the plurality of corresponding connection requests;

establishing a connection socket for at least one of the plurality of connection requests without placing the connection request in a backlog queue;

for each connection request for which the server computer system cannot currently establish a connection socket, placing the connection request in the backlog queue without then establishing a connection socket; and

identifying and disconnecting one or more of the connection sockets that are suspected to be serving a malicious connection request and that have not received associated request data for the one or more corresponding connection requests, so as to reduce any affect of the denial of service attack.

27.     (New) A method as recited in claim 26, wherein the one or more identified connection sockets are disconnected only upon determining that the backlog queue is being used.

28.     (New) A method as recited in claim 27, wherein the one or more identified connection sockets are disconnected only after waiting a predetermined period of time after determining that the backlog queue is being used.

29.     (New) A method as recited in claim 1, wherein the backlog queue is capable of containing connection requests that include associated request data and connection requests that do not include associated request data

30.     (New) A method as recited in claim 1, wherein identifying connection sockets that have no received request data includes identifying connection sockets that are suspected to be serving a malicious connection request.

31.     (New) A computer program product as recited in claim 17, wherein the backlog queue is capable of containing connection requests that include associated request data and connection requests that do not include associated request data

32.     (New) A computer program product as recited in claim 17, wherein identifying connection sockets that have no received request data includes identifying connection sockets that are suspected to be serving a malicious connection request.